

What is claimed is:

1 1. A method of handling operation in a multi-threaded processing system, comprising:

2 determining if a stalled operation of a first thread is due to a loading of data from a

3 memory device; and

4 flushing an instruction from said first thread from a pipeline of said processing system

5 when data is to be loaded from said memory device before executing said instruction.

1 2. The method of claim 1 wherein said memory device is system memory coupled to a

2 memory bus.

1 3. The method of claim 1 further comprising:

2 marking said instruction as a miss.

1 4. The method of claim 3 further comprising:

2 rescheduling said instruction to be executed in said pipeline.

1 5. A method of handling operation in a multi-threaded processing system, comprising:

2 determining if a stalled operation of a first thread is due to a loading of data from a

3 memory device; and

4 flushing an instruction from said first thread from a pipeline of said processing system

5 when data is to be loaded after a predetermined number of clock cycles from said memory device

6 before said instruction can be executed.

1 6. The method of claim 5 wherein said memory device is system memory coupled to a
2 memory bus.

1 7. The method of claim 6 further comprising:
2 marking said instruction as a miss.

1 8. The method of claim 7 further comprising:
2 rescheduling said instruction to be executed in said pipeline.

1 9. The method of claim 8 further comprising:
2 executing said instruction when data is loaded from said memory device.

1 10. A processing system comprising:
2 a scheduler to pass instructions from first thread and second threads to an execution
3 pipeline; and
4 pipeline control logic coupled to said execution pipeline to determine if a stalled
5 execution of a first thread is due to a loading of data from a memory device and to flush an
6 instruction from said first thread from said execution pipeline when data is to be loaded from
7 said memory device before said instruction can be executed.

1 11. The processing system of claim 10 wherein said pipeline control logic is to mark said
2 instruction as a miss.

1 12. The processing system of claim 10 further comprising:
2 an exception and retirement logic coupled to said execution pipeline.

1 13. The processing system of claim 12 wherein said instruction marked as a miss is to be
2 detected by said exception and retirement logic.

1 14. The processing system of claim 13 further comprising:
2 a fetch unit to provide said instruction to said scheduler.

1 15. The processing system of claim 14 wherein said pipeline control logic is to cause said
2 instruction to be executed when data is loaded from said memory device.

1 16. A computing system comprising:
2 a memory bus coupled to system memory; and
3 a processing system coupled to said memory bus, said processing system including
4 a scheduler to pass instructions from first thread and second threads to an
5 execution pipeline; and
6 pipeline control logic coupled to said execution pipeline to determine if a stalled
7 execution of a first thread is due to a loading of data from system memory and to flush an
8 instruction from said first thread from said execution pipeline when data is to be loaded
9 from said system memory before said instruction can be executed.

1 17. The computing system of claim 16 wherein said pipeline control logic is to mark said
2 instruction as a miss.

1 18. The computing system of claim 10 wherein said processing system further includes
2 an exception and retirement logic coupled to said execution pipeline.

1 19. The computing system of claim 18 wherein said instruction marked as a miss is to be
2 detected by said exception and retirement logic.

1 20. The computing system of claim 19 wherein said processing system further includes
2 a fetch unit to provide said instruction to said scheduler.

1 21. The computing system of claim 20 wherein said pipeline control logic is to cause said
2 instruction to be executed when data is loaded from said system memory.